

AD-A146 288

DESIGNING OPTIMAL AOQL SAMPLING PLANS A COMPUTERIZED
APPROACH(U) FLORIDA UNIV GAINESVILLE DEPT OF INDUSTRIAL
AND SYSTEMS ENGINEERING J E WALKER ET AL JAN 84

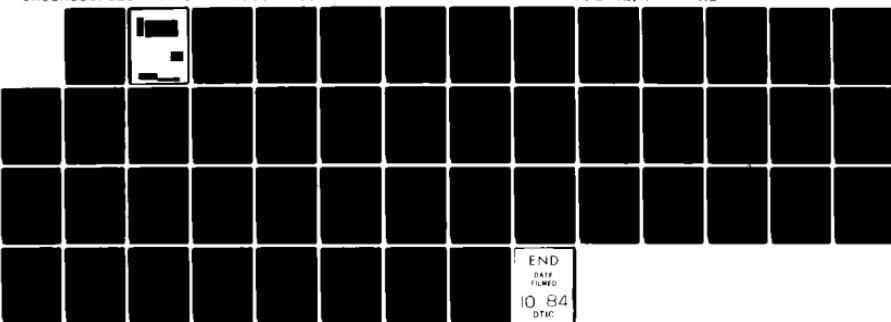
1/1

UNCLASSIFIED

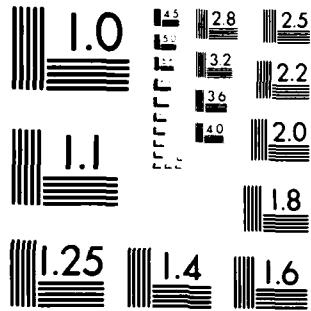
RR-84-1 N00014-83-K-0093

F/G 12/1

NL

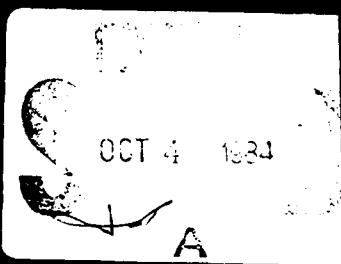


END
DATE
FILED
10 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AD-A146 288



This document contains neither recommendations nor conclusions of the Defense Intelligence Agency. It has been reviewed and approved for public release by the Defense Intelligence Agency.

84 09 11 063

DESIGNING OPTIMAL AOQL SAMPLING PLANS
A COMPUTERIZED APPROACH

Research Report No. 84-1

by

Jo Ellen Walker
and
Richard S. Leavenworth

January 1984

Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida 32611

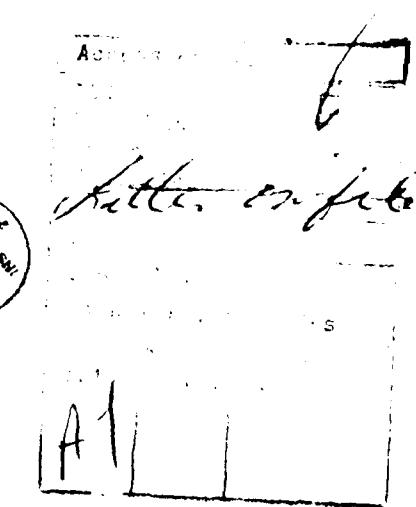
This research was supported by the U.S. Department of the Navy,
Office of Naval Research, Under Contract N0014-83-K-0093.

FINDINGS OF THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL
DEPARTMENT OF THE NAVY POSITION UNLESS SO DESIGNATED BY OTHER
AUTHORIZED DOCUMENTS.

This document contains neither recommendations nor conclusions of the National Institute of Standards and Technology. It is the property of NIST and is loaned to your agency; it and its contents are not to be distributed outside your agency without the permission of the Director of NIST.

TABLE OF CONTENTS

	PAGE
ABSTRACT	i
OBJECTIVE	1
PROBLEM FORMULATION	5
ALGORITHM	7
ANALYSIS OF THE ALGORITHM	10
COMPARATIVE RESULTS	25
REFERENCES	29
APPENDIX	30



ABSTRACT

This study provides an algorithm and computer program, coded in Fortran IV, to develop double sampling acceptance plans for attributes which satisfy a specified Average Outgoing Quality Limit (AOQL) and minimize the Average Fraction Inspected (AFI) at a specified quality level, $P_{l,0}$. An analysis of the steps in the algorithm is provided and comparison is made among the results of program runs and Dodge-Romig AOQL and MIL-STD-105D plans.

OBJECTIVE

This study develops a computer based algorithm to derive double sampling plans which minimize the Average Fraction Inspected, AFI(p_1), at a designated quality level, p_1 . The AFI is defined as:

$$AFI = 1/N ((L(p)*n_s) + N*(1-L(p))) : \quad \text{Single Sampling (1)}$$

and,

$$AFI = 1/N (L(p)*n_1) + Pa(n_2)*n_2 + N*(1-L(p)); \quad \text{Double Sampling (2)}$$

$L(p)$ is the appropriate formulation for the probability of acceptance as a function of the incoming quality level, p . This study employs the binomial distribution which, for the single sampling case, yields:

$$L(p) = \sum_{d=0}^c \binom{n}{d} p^d (1-p)^{n-d} \quad (3)$$

where n is the sample size and c the acceptance number.

For Double Sampling,

$$L(p) = \sum_{d_1=0}^{c_1} \binom{n_1}{d_1} p^{d_1} (1-p)^{n_1-d_1} + \sum_{d_1=c_1+1}^{r_1-1} \sum_{d_2=0}^{c_2-d_1} \binom{n_1}{d_1} \binom{n_2}{d_2} p^{d_1+d_2} (1-p)^{n_1+n_2-d_1-d_2} \quad (4)$$

where n_1 is the first sample size of a double sampling plan, n_2 is the second sample size, c_1 , and c_2 are the acceptance numbers on the first and second samples, respectively, and r_1 is the rejection number of the first sample.

The rejection number on the second sample, r_2 , always is c_2+1 .

$P_a(n_1)$ is defined as the probability of acceptance on n_1 given as

$$P_a(n_1) = \sum_{d_1=0}^{c_1} \binom{n_1}{d_1} p^{d_1} (1-p)^{n_1-d_1} \quad (5)$$

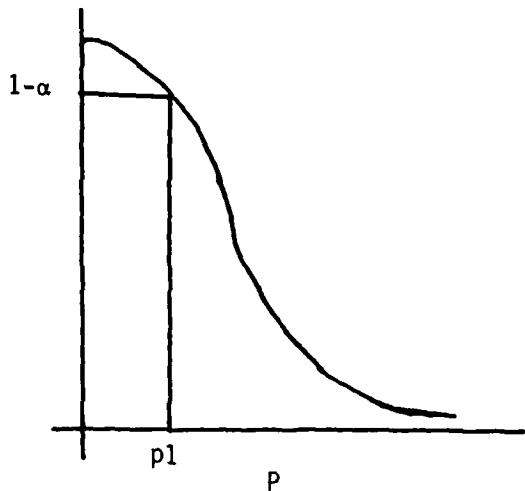
$P_a(n_2)$ is the probability of acceptance on n_2 given as

$$P_a(n_2) = \sum_{d_1=c_1+1}^{r_1-1} \sum_{d_2=0}^{c_2-d_1} \binom{n_1}{d_1} \binom{n_2}{d_2} p^{d_1+d_2} (1-p)^{n_1+n_2-d_1-d_2} \quad (6)$$

Feasible sampling plans are those which strictly satisfy:

$$L(p_1) > 1-\alpha \quad (7)$$

where p_1 is the process average or the quality level considered acceptable for the purpose of acceptance sampling. The probability of acceptance must be least $1-\alpha$, where α is defined as the Producer's Risk. (p_1 and α are specified design parameter).

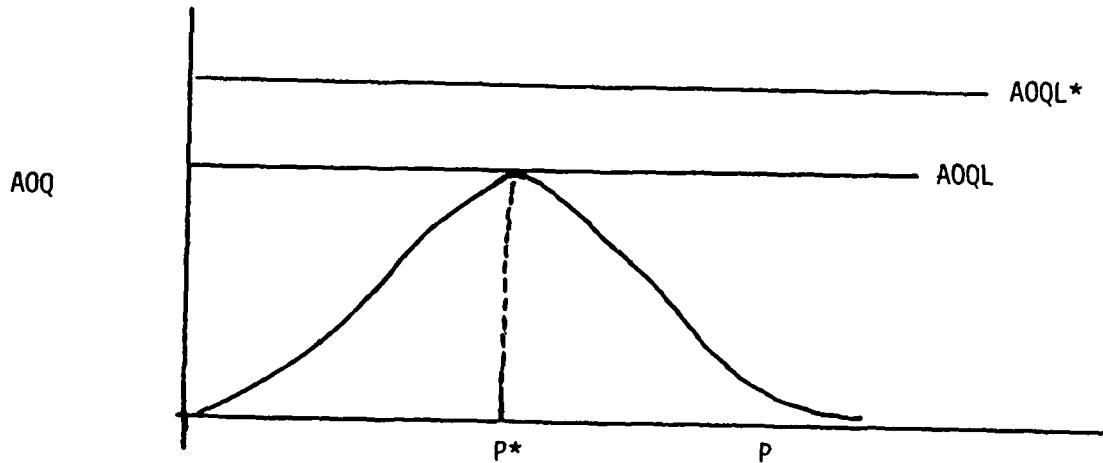


and $\text{AOQL} < \text{AOQL}^*$ (8)

where AOQL^* : specified Average Outgoing Quality Limit

and AOQL : $\text{AOQ}(p^*) = p^*(1-\text{AFI}(p^*))$ for a given sampling plan

p^* : p value at which the AOQ function reaches its maximum value



It is useful at this point to differentiate between design parameters and acceptance sampling plan parameters. The design parameters for either a single sampling or double sampling plan are the required input values, p_1

and α , the so-called Producer's Risk Point, and the required maximum AOQL, AOQL*. The plan parameters are the outputs of the programmed algorithm, n_s and c for a single sampling plan and n_1 , n_2 , c_1 , c_2 , and r_1 , for a double sampling plan.

PROBLEM FORMULATION

Initially, a single sampling plan is found such that the AFI(p_1) is minimized. The minimum (n_s, c) combination satisfying $AOQL < AOQL^*$ and satisfying $L(p_1) > 1-\alpha$ yields the optimal single sampling plan which is designated (n^*, c^*) .

The bounds on the parameters of the double sampling plan are based on these results from deriving the single sampling plan. The first sample, n_1 , must satisfy

$$c_2 < n_1 < n^* \quad (9)$$

$$c_2 > c^* \quad (10)$$

$$c_1 < c_2-1 \quad (11)$$

$$\text{and } B(n_1, c_2, p_1) > 1-\alpha \quad (12)$$

where c_1 and c_2 are the acceptance numbers of the first and second samples, respectively, of a double sampling plan, and $B(n, c, p)$ is the cumulative probability for a binomial distribution as given in equation (3). For purposes of this study, r_1 has been made equal to r_2 , that is, c_2+1 , following the Dodge-Romig schemes. The second sample, n_2 , must satisfy:

$$n_2 > n^*-n_1 \quad (13)$$

and

$$BB(n_1, n_2, c_1, c_2, p_1) > 1-\alpha \quad (14)$$

where $BB(n_1, n_2, c_1, c_2, p)$ is the double cumulative probability of a binomial distribution as given in equation (4).

The objective is to minimize AFI(p_1) subject to:

$$L(p_1) > 1-\alpha$$

$$AOQL < AOQL^*$$

For each feasible c_1, c_2 combination, an optimal n_1, n_2 combination is found such that the $AFI(p_1)$ is minimized. The minimum AFI from all c_1, c_2 combinations yields the optimal sampling plan, $n_1^*, n_2^*, c_1^*, c_2^*$.

ALGOROTHM

The algorithm is divided into two parts, one for single sampling and one for double sampling. The bounds on some parameters of the double sampling plan are based on the results of deriving a corresponding single sampling plan, initially the optimal single sampling plan. Thus, to simplify discussion of the algorithm, the two parts are treated separately.

Deriving the single sampling plan, (ns,c).

1. Initialization: $ns=1$, $c=0$. Input α (> 0), p_1 , $AOQL^*$ ($> p_1$).
2. Compute $B(ns,c,p_1)$.
3. Test: Is $B(ns,c,p_1) > 1-\alpha$?
 - (a) If so, go to step 4.
 - (b) If not so, increment c , set $ns=c+1$, and go to step 2.
(If the first value of $B(ns,c,p_1)$ is not $> 1-\alpha$, then no larger value of ns can satisfy the constraint.)
4. Compute the AOQL for ns,c .
5. Test: Is the AOQL $< AOQL^*$?
 - (a) If so, go to step 6.
 - (b) If not so, increment ns and go to step 4.
 $(c < ns < N)$
6. Compute $B(ns,c,p_1)$.
7. Test: Is $B(ns,c,p_1) > 1-\alpha$?
 - (a) If so, go to step 8.
 - (b) If not so, increment c , set $ns=c+1$, and go to step 4.
8. Compute the AFI(p_1). Program outputs ns and c along with the AFI(p_1) and actual AOQL after double sampling algorithm.

9. Set $n^*=ns$, $c^*=c$, go to double sampling algorithm. (Note - in future cycles step 8 is not employed.)

Deriving the Double Sampling Plan (n_1, n_2, c_1, c_2)

1. Initialization: Set $c_2=c^*$, $c_1=0$, $n_1=c_2+1$, and $n_2=n^*-n_1$; go to step 4.
($B(n_1=c_2+1, c_2, p_1) > 1-\alpha$ is always satisfied.)
2. Compute $B(n_1, c_2, p_1)$.
3. Test: Is $B(n_1, c_2, p_1) > 1-\alpha$?
 - (a) If so, go to step 4.
 - (b) If not so, increment c_2 , set $c=c_2$, set $n=n^*$, go to step 2 of single sampling algorithm. (If the first value of $B(n_1, c_2, p_1)$ is not $> 1-\alpha$, then no larger value of n_1 can satisfy the constraint.)
4. Compute $BB(n_1, n_2, c_1, c_2, p_1)$
5. Test: Is $BB(n_1, n_2, c_1, c_2, p_1) > 1-\alpha$?
 - (a) If so, go to step 6
 - (b) If not so, increment c_1 and go to step 4. (No larger value of n_2 will satisfy the constraint.)
6. Compute the AOQL for the plan (n_1, n_2, c_1, c_2)
7. Test: $AOQL(n_1, n_2, c_1, c_2) < AOQL^*$
 - (a) If so, go to step 8
 - (b) If not so, increment n_2 and go to step 6. (AOQL values decrease with increasing values of n_2 .)
8. Compute $BB(n_1, n_2, c_1, c_2, p_1)$
9. Test: Is $BB(n_1, n_2, c_1, c_2, p_1) > 1-\alpha$?
 - (a) If so, go to step 10.
 - (b) If not so, go to step 9a.

9a. Test: Does $n_1=n^*$?

- (a) If so, increment c_1 , set $n_1=c_2+1$, $n_2=n^*-n_1$, and go to step 2.
- (b) If not so, increment n_1 , set $n_2=n^*-n_1$, and go to step 2.

10. Compute $AFI(n_1,n_2,c_1,c_2,p_1)$.

11. Test: is $AFI(\text{current plan}) < AFI(\text{previous plan})$?

- (a) If so, store AFI (current plan) and its plan parameters for the set value of c_2 and go to step 12.
- (b) If not so, retain AFI (previous plan) and its plan parameters for the set value of c_2 and go to step 13.

12. Test: Does $n_1=n^*$?

- (a) If so, increment c_1 , set $n_1=c_2+1$, $n_2=n^*-n_1$, and go to step 2.
- (b) If not so, increment n_1 , set $n_2=n^*-n_1$, and go to step 2.

13. Test: Does $c_1=c_2-1$?

- (a) If so, go to step 14.
- (b) If not so, increment c_1 , set $n_1=c_2+1$, $n_2=n^*-n_1$, and go to step 2.

14. Test: Is $AFI(n_1,n_2,c_1,c_2,p_1) < AFI(n_1,n_2,c_1,c_2-1,p_1)$?

- (a) If so, retain AFI (current plan) and its plan parameters, increment c_2 , set $c=c_2$, $n=n^*$, and go to step 2 of the single sampling algorithm.
- (b) If not so, previous plan AFI and its parameters (n_1,n_2,c_1,c_2) is optimal.

ANALYSIS OF THE ALGORITHM

The $L(p_1)$ decreases with increasing values of n . This result reduces the computational time greatly. As n_1 increases, $B(n_1 - 2, p_1)$ decreases (Figure 1). Once the $L(p_1)$ constraint is violated, c_2 is incremented rather than computing $B(n_1, c_2, p_1)$ for all $n_1 < n^*$ for the given c_2 . Also, since the double cumulative probabilities decrease with increasing values of n_2 , the next n_1 value is generated once $BB(n_1, n_2, c_1, c_2, p_1) < 1-\alpha$ instead of continuing the computation for $n_2 < N-n_1$ (Figure 2).

Table 1 perhaps best simplifies further discussion. The tabular results are from a computer run wherein the specified parameters are $p_1=0.03$, $\alpha=0.05$, $AOQL=0.05$, and $N=500$. Each cell (block) contains the (n_1, n_2) pair and corresponding AFI(p_1) values such that $L(p_1) > 1-\alpha$ and $AOQL < AOQL^*$ for each feasible (c_1, c_2) combination, i.e. for $c_1 < c_2$. Table 2 shows the optimal plan within each cell (c_1, c_2 combination), the column minimum (fixed c_2), and the optimal plan selected from the column minimums.

By studying the results as displayed in Table 1, the following conclusions are reached:

1. In each cell, the AFI(p_1) increases with increasing n_2 and fixed n_1 (Figure 3). Therefore, it is necessary to find only the first n_1, n_2 combination satisfying the constraints rather than testing all feasible values of n_2 . The large list of the AFI values for all values of n_2 and a given n_1 is not included.
2. In each cell, the AFI is a convex function of each minimum n_1, n_2 combination (first feasible n_1, n_2 combination). The minimum sampling plan for each cell is one associated with the minimum of this function (Figure 4.)
- 4.) In Table 1, the minimum sampling plan of the first cell in the $c_2=2$ column is $n_1=11$, $n_2=16$.

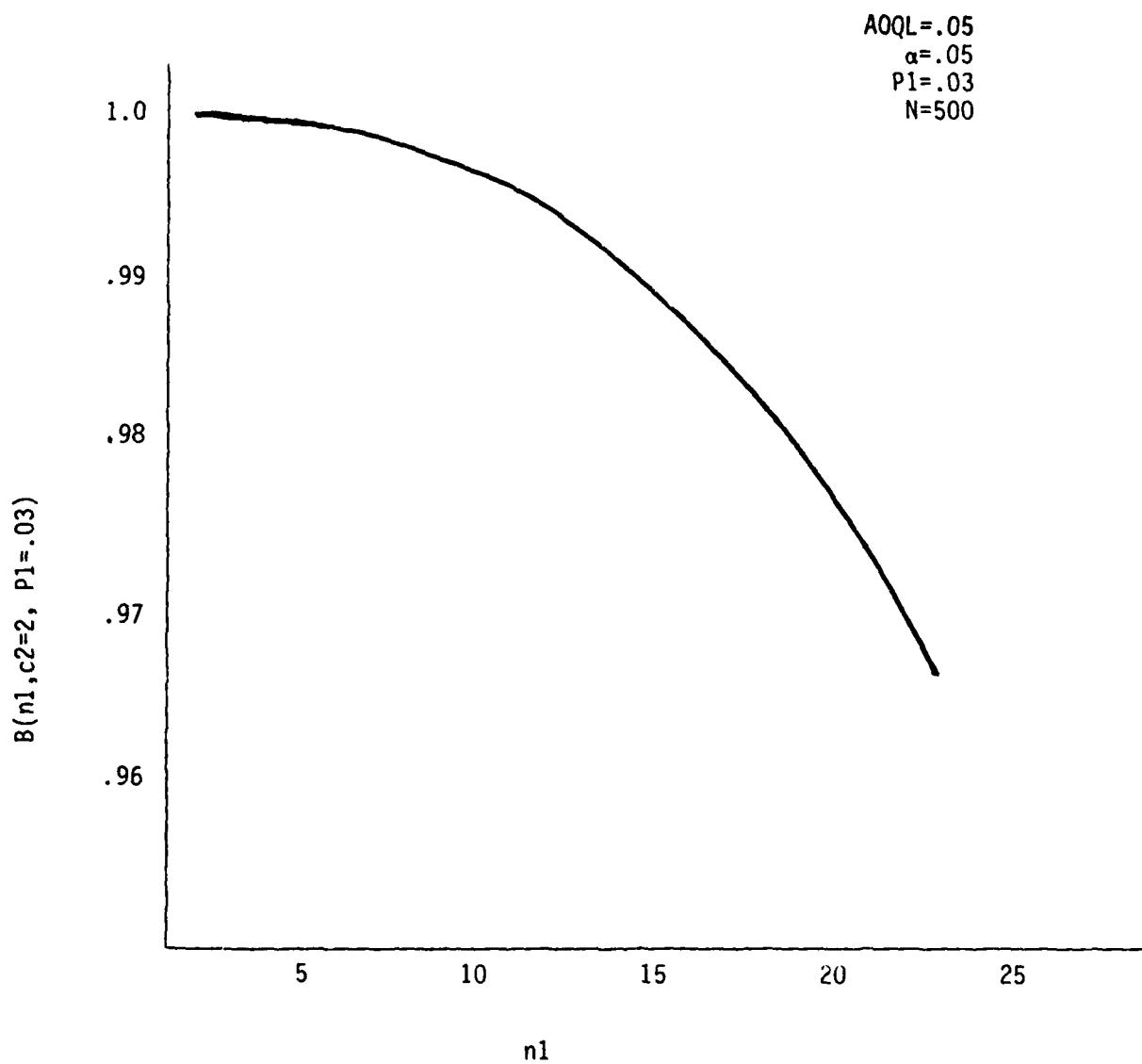


FIGURE 1. Binomial Probabilities as a Function of First Sample Size, n₁.

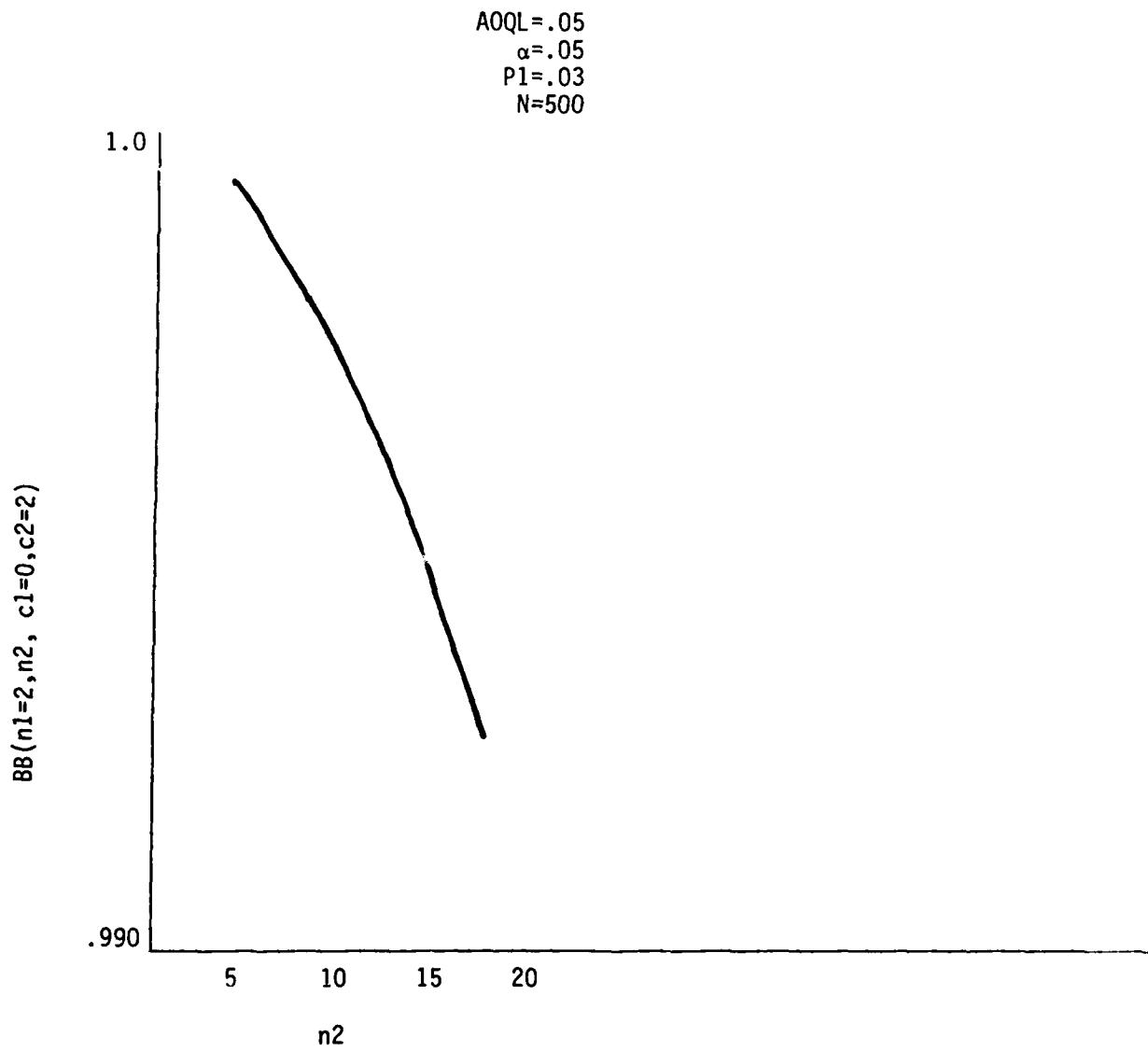


FIGURE 2. Double Binomial Probabilities as a Function of the Second Sample Size, n_2 .

TABLE 1
Schematic of Program Outputs, Double Sampling

	c2=2			c2=3			c2=4			c2=5		
	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
c1=0	9	23	.073063	7	54	.076090	7	67	.065837	7	80	.061384
	10	20	.072404	8	41	.062998	8	54	.057350	8	66	.055319
	11	16	.067187	9	35	.059519	9	47	.054957	9	58	.053936
	12	14	.067326	10	30	.056563	10	41	.052956	10	52	.053722
	13	13	.07038	11	26	.054611	11	37	.052887	11	47	.053912
	14	12	.073129	12	25	.057869	12	35	.054955	12	45	.056551

	20	16	.085011	31	6	.091863	41	5	.102421	51	6	.17795 cell
c1=1	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
	18	15	.082664	17	38	.071300	16	94	.083959	15	110	.077789
	19	12	.084097	18	30	.068552	17	53	.063678	17	67	.058314
	20	9	.084268	19	24	.066407	18	43	.060595	18	56	.057129
	21	7	.085656	20	21	.067226	19	37	.059908	19	49	.056860
	22	5	.086196	21	17	.066042	20	32	.059564	20	48	.057501
	23	4	.089133	22	15	.067319	21	28	.059740	21	39	.057907

	26	0	.091797	31	6	.087214	41	6	.098053	47	10	.108607
c1=2	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)	n1	n2	AFI(p1)
	26	48	.084916	26	70	.080637	26	89	.077190			
	27	23	.080726	27	41	.074589	27	57	.070797			
	28	17	.081398	28	32	.073855	28	47	.070319			
	29	13	.082497	29	26	.073923	29	40	.070588			
	30	10	.083788	30	22	.074750	30	34	.070989			
			
	33	4	.088216	41	6	.095250	51	6	.110683			
c1=3				n1	n2	AFI(p1)	n1	n2	AFI(p1)			
				37	27	.089033	37	46	.085970			
				38	19	.090008	38	45	.086241			
				39	14	.091277	39	29	.087349			
				40	11	.093011	40	24	.088489			
				41	8	.094413	41	20	.089759			
						
				46	1	.103962	51	6	.109071			
c1=4							n1	n2	AFI(p1)			
							47	32	.102570			
							48	21	.103790			
							49	15	.105285			
							50	12	.107183			
							51	9	.108927			

The minimums for each cell are given in Table 2.

TABLE 2

n_1	n_2	$AFI(p_1)$									
11	16	.067187	11	26	.054611	11	37	.052887	10	52	.053722
18	15	.082664	21	17	.066042	20	32	.059564	19	49	.056860
			27	23	.080726	28	32	.073855	28	47	.070319
						37	27	.089033	37	46	.085970
									47	32	.102570

Column Minimum

$n_1=11$ $c_1=0$
 $n_2=16$ $c_2=2$
 $AFI=.067187$

Column Minimum

$n_1=11$ $c_1=0$
 $n_2=26$ $c_2=3$
 $AFI=.054611$

Column Minimum

$n_1=11$ $c_1=0$
 $n_2=37$ $c_2=4$
 $AFI=.052887$

Column Minimum

$n_1=10$ $c_1=0$
 $n_2=52$ $c_2=5$
 $AFI=.053722$

Minimum double sampling plan: $c_1=0$, $c_2=4$, $n_1=11$, $n_2=37$, $AFI(p_1)=0.052887$
 $AQQL=0.49464$

Corresponding single sampling plan: $c=2$, $n=26$, $AFI(p_1)=0.091796$, $AQQL=0.049770$

$\alpha = .05$ $C_1 = 0$
 $P_1 = .03$ $C_2 = 3$
 $AOQL = .05$ $N_1 = 12$
 $N = 500$

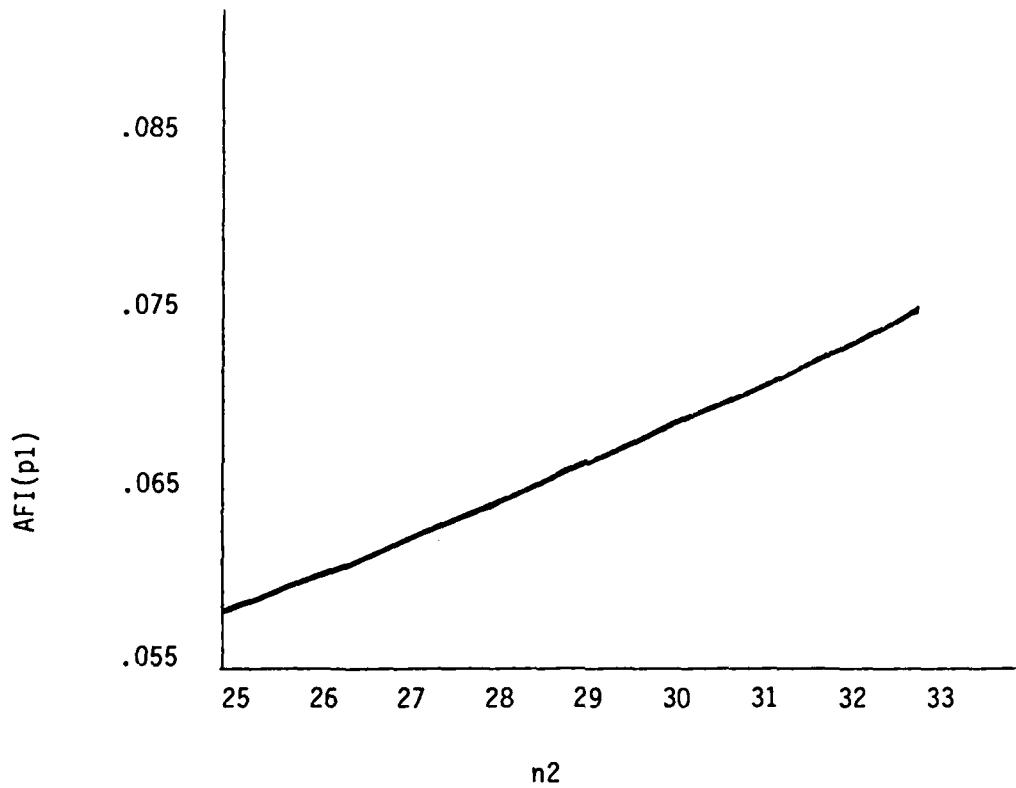


FIGURE 3. Change in AFI(p_1) as a Function
of n_2 .

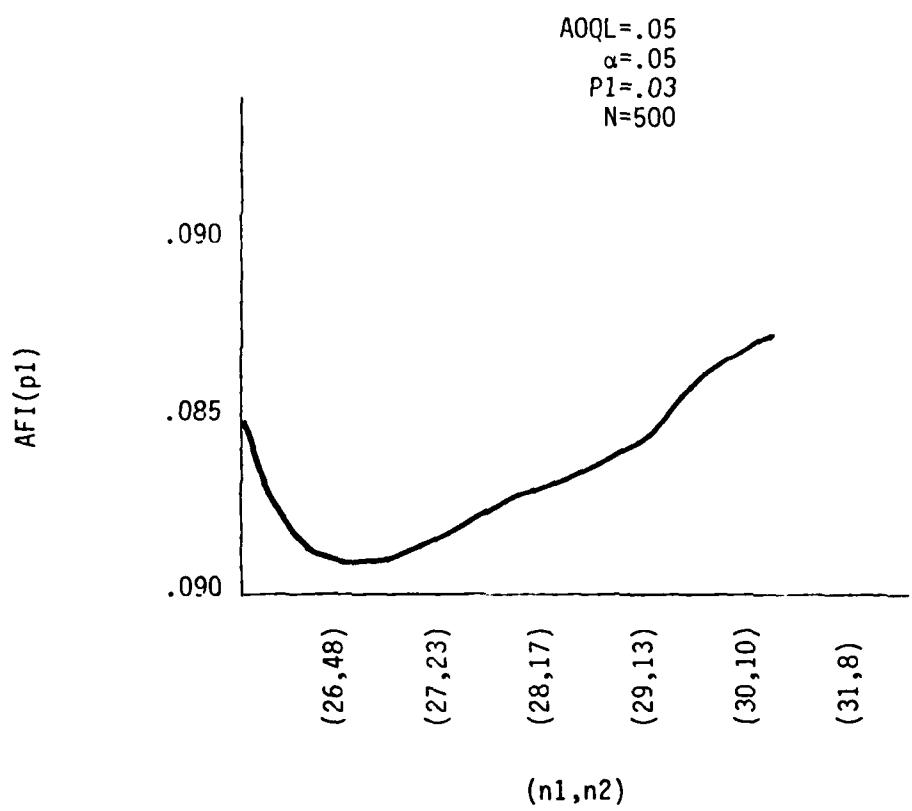


FIGURE 4. $AFI(p_1)$ as a Function of
c₁, c₂ Cell Minimums

3. Each column gives cells for all feasible c_1 values for a given c_2 . The cell AFI minimums are also convex in c_1 . Therefore, the minimum sampling plan for a given c_2 is the one corresponding to the minimum of these cell minimums (Figure 5). For example, the minimum AFI of column 2 is 0.054611 and the corresponding sampling plan is $c_1=0$, $c_2=3$, $n_1=11$, $n_2=26$.
(Note: It is not necessarily the case that the minimum always will occur at the lowest feasible value of c_1 . For example, for the parameters $\alpha=.023606$, $p_1=.05$, $AOQL^*=.05$, and $N=350$, the minimum sampling plan is $c_1=1$, $c_2=8$, $n_1=21$, $n_2=62$. However, the minimum feasible value of c_1 is 0.)
4. The minimums of the columns also follow this convex pattern. The column minimum AFI(p_1) values in Table 1 corresponding to $c_2=2,3,4,5$ are 0.072404, 0.0544611, 0.052887, 0.053722, respectively. This provides the minimum samling plan as well as a stopping criterion for the algorithm (Figure 6). Thus the minimum of the function, 0.052887, corresponds to the minimum sampling plan, $c_1=0$, $c_2=4$, $n_1=11$, $n_2=37$.

Single Sampling

The first feasible ns,c combination is found such that $B(ns,c,p_1) > 1-\alpha$. Since the binomial probabilities decrease as ns increases, the first infeasible value of ns indicates that all probabilities associated with larger values of ns for a given c are $< 1-\alpha$. Thus, once a point becomes infeasible, c is incremented, and the process begins again.

For the first feasible ns value, the $AOQL < AOQL^*$ constraint is checked. This is accomplished in a subroutine which utilizes the Golden Section Method [3] to find the value of p,p^* , which maximizes the AOQ

$AQCL = .05$
 $\alpha = .05$ $C2 = 4$
 $P1 = .03$
 $N = 500$

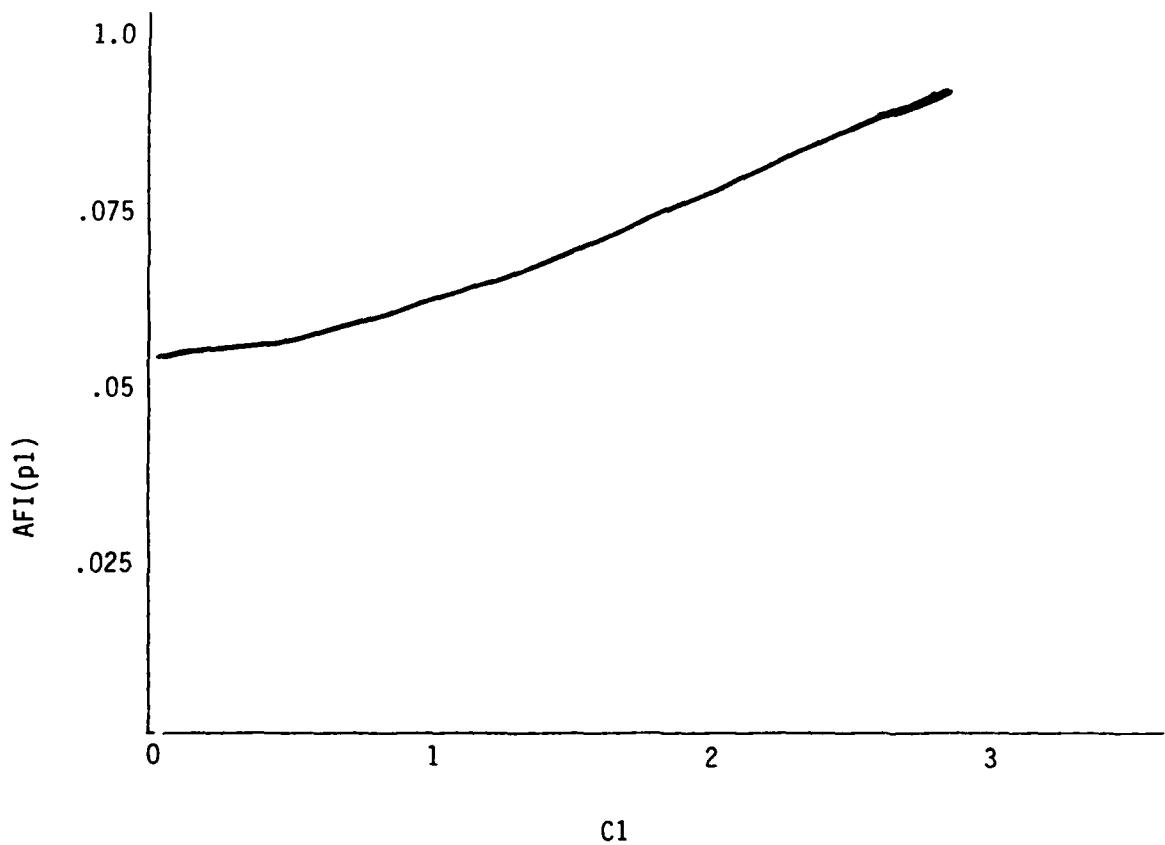


FIGURE 5. Minimum Cell AFI(p_1) Values
as a Function of c_1 .

AOQL=.05
 $\alpha=.05$
 $P_1=.05$
 $N=500$

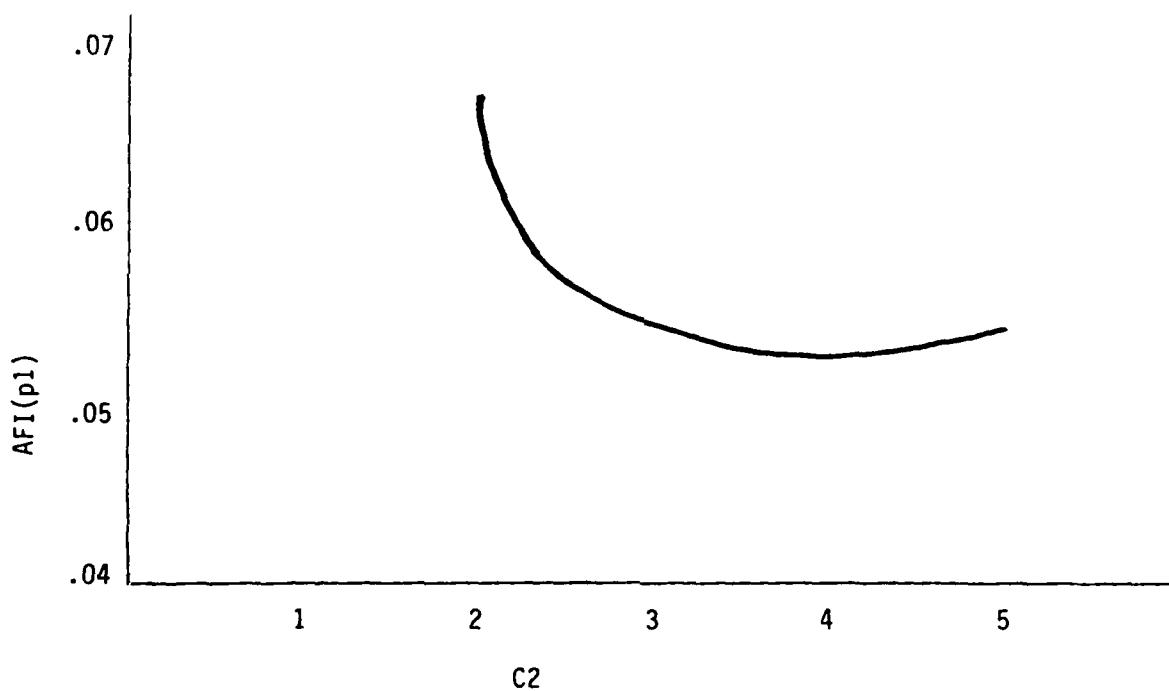


FIGURE 6. Value of AFI(p_1) as a Function of c_2 .

function, i.e., the AOQL. p^* may fall in the interval (0,1). The calculation of the required initial left bound (PLFT) and the initial right bound (PRT) are based on this allowable interval. In each iteration of the search, the AOQ values associated with PLFT and PRT are computed. If $\text{AOQ}(\text{PLFT}) \geq \text{AOQ}(\text{PRT})$, $\text{PRT}=\text{PLFT}$. If not, $\text{PLFT}=\text{PRT}$, i.e., after comparing the AOQ values, only one of the p values is new. Thus only one additional computation of the AOQ is necessary. When the interval on p is less than 0.0001, the search ends, and for the necessary accuracy, it is assumed that $p^*=(\text{PLFT}+\text{PRT})/2$. If $\text{AOQL} > \text{AOQL}^*$, the AFI is computed and the process begins again.

If $\text{AOQL} < \text{AOQL}^*$, a bisection method is used to find the first ns value that satisfies the AOQ constraint. The initial bounds for the bisection are ns and N . If the AOQL value corresponding to N is greater than AOQL^* , the bisection is omitted, c is incremented, and the process begins again. This is allowable since the AOQL values decrease with increasing values of ns (Figure 7). Use of the bisection method in this case appears to be very efficient. It requires $\log(2)N$ iterations rather than the $N-ns$ possible iterations needed using a total enumeration method.

If the associated binomial probabilities are $> 1-\alpha$, the $\text{AFI}(p_1)$ is computed. Otherwise, c is incremented, ns is set to the new c value, and the process begins again.

Double Sampling

The double sampling procedure is very similar. The size of the first sample, n_1 , ranges from c_2+1 to n^* . If $B(n_1, c_2, p_1) < 1-\alpha$, c_2 is incremented. Since these cumulative binomial probabilities decrease with increasing n_1 , further enumeration for a given c_2 is unnecessary. Whenever c_2 is incremented, new bounds on n^* must be computed using the single sampling algorithm. Since c_2 corresponds to c^* , a new n^* satisfying the single

AOQL=.05
 $\alpha=.05$
 $P_1=.03$
 $N=500$

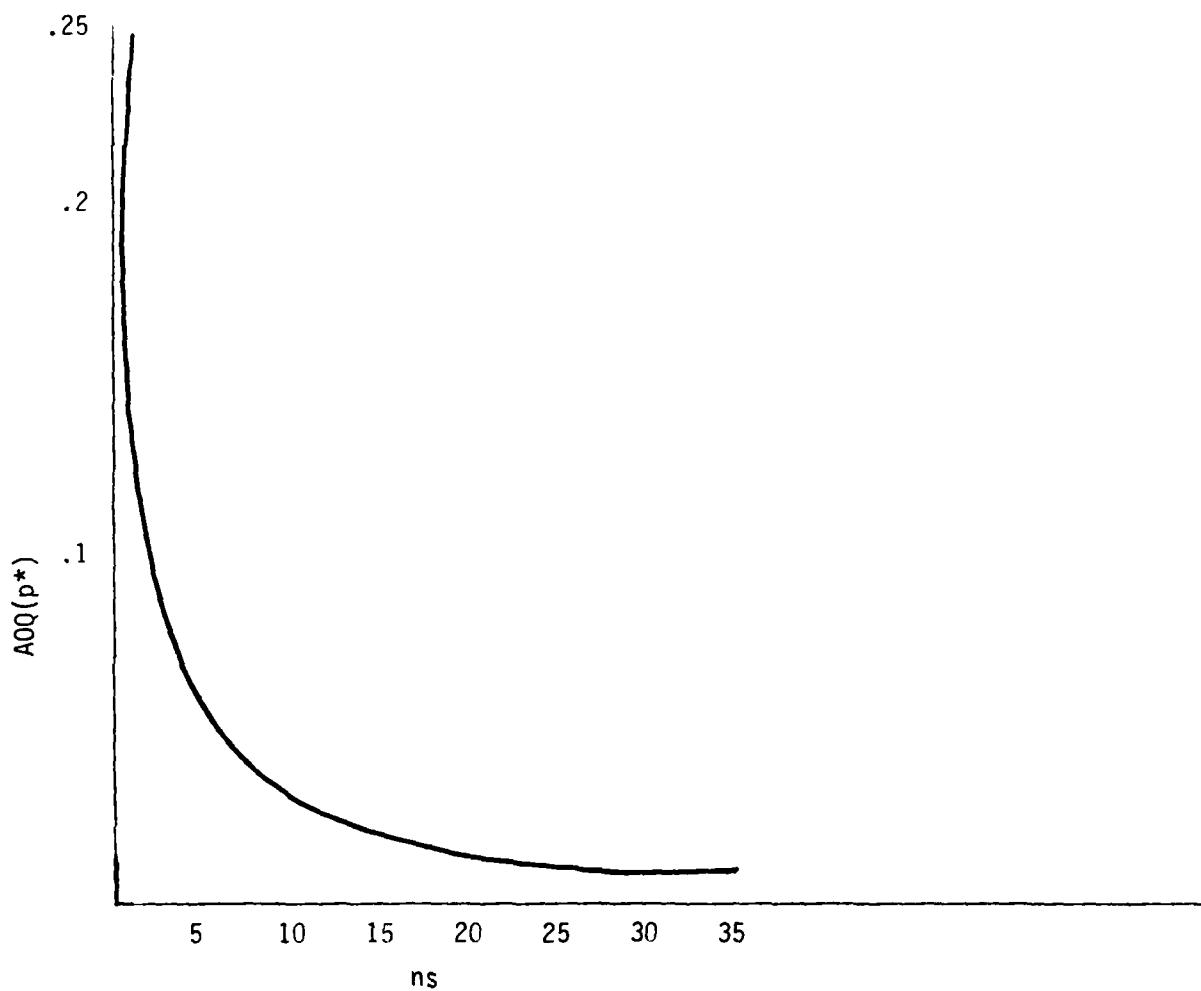


FIGURE 7. Values of AOQ as a Function of ns.

sampling criteria must be found. The double sampling algorithm continues with these new values of $c_2=c^*$, $c_2 < n_1 < n^*$, and $n_2=n^*-n_1$.

If $B(n_1, c_2, p_1) > 1-\alpha$, the corresponding feasible $n_2 > n^*-n_1$ is found. If $BB(n_1, n_2, c_1, c_2, p_1) < 1-\alpha$, c_1 is incremented since the double cumulative probabilities decrease for the first feasible n_1, n_2 combinations of a given c_1 (Figure 8). Otherwise, a bisection technique similar to that used for the single sampling case is used to find the lowest n_2 value satisfying $AQQL < AQQL^*$. If the cumulative double probability is greater than or equal to $1-\alpha$, the $AFI(p_1)$ is computed. If not, n_1 is incremented and the process begins again. For a fixed n_1 , the first feasible (n_1, n_2) pair is the minimum of the AFI function.

The minimum sampling plan is one such that the $AFI(p_1)$ is minimized. First, the minimum for each c_1 value, the cell minimum, is determined. This consists of the n_1, n_2 combination which yields the smallest AFI . As previously stated, the function is convex, however there is some variation since n_2 does not decrease in uniform increments as n_1 increases. In Table 1, when $c_1=1$, $c_2=3$, the AFI values decrease to 0.066407, increase to 0.067226, decrease to 0.066042, and finally increase to 0.067319 as n_1 was increased. If the function were strictly convex, the computation for each cell would end upon the first increase in the value of $AFI(p_1)$. To account for this discontinuity, the algorithm continues for five additional increments of n_1 to reasonably insure that a global minimum $AFI(p_1)$ is reached.

Next, the minimum for all feasible c_1 values, the column minimum, is found. Due to the convexity of the AFI values corresponding to the c_1 values, a column minimum is found after the first cell minimum increases.

The last step is to find the minimum AFI of all columns. The algorithm terminates after the minimum AFI corresponding to a c_2 value, (column

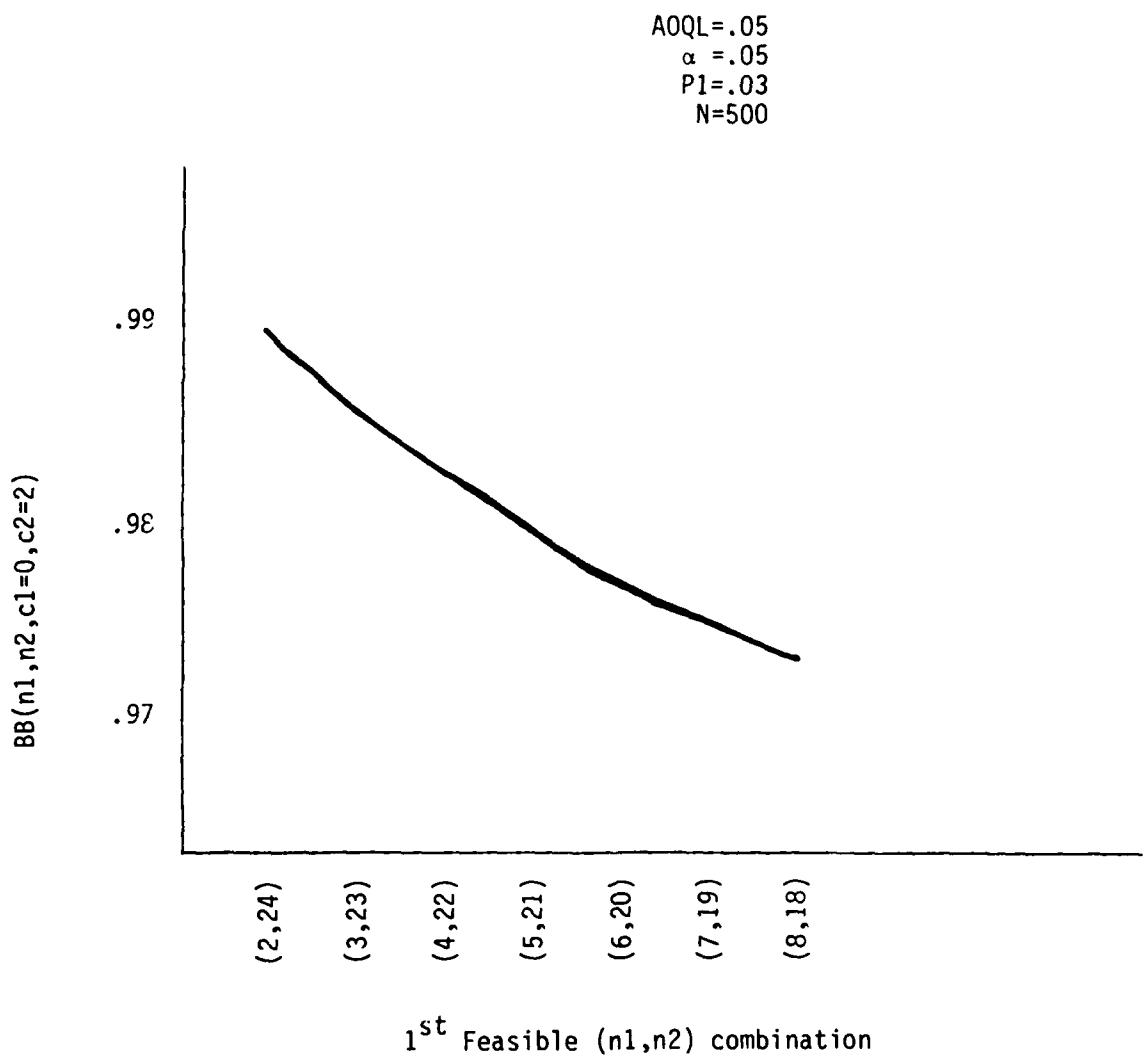


FIGURE 8. Values of $L(p)$ for First Feasible n_1, n_2 Combinations

minimum), is larger than the preceding one. Thus, the minimum sampling plan for the design parameters is found.

The computer code was written in Fortran IV and run on the PDP 11/34 digital computer. For further clarification of the computer code, flowcharts of the main program and the subroutines that compute p^* and AOQL are included in the appendix as well as a listing of the code.

COMPARATIVE RESULTS

The computer program was run with various AOQL values. The following table characterizes the effectiveness of the program. Sampling plans derived from a study that minimizes the ASN(p_1), plans derived by Dodge and Romig, and MIL-STD-105D plans are shown for comparison.

The values of n_1 and n_2 given in the Dodge Romig tables [1] are based on the largest lot size in each range. The values of c_1 and c_2 correspond to the mean lot size and to the mean value of the process average. This ensures that the AOQL values over the range will not exceed the specified value, however this only gives the average most economical plan within a range of process averages and lot sizes. The computed sampling plan will vary as the lot size and design process average change. For example, for AOQL=1%, $\alpha=0.03465$, $p_1=0.0065$, and a lot size of 301 (lower bound of the Dodge-Romig range), the computer program gives a sampling plan of $c_1=0$, $c_2=2$, $n_1=40$, and $n_2=84$ with an AFI(p_1)=0.2170. At $N=400$, the upper bound of the lot size range, $c_1=0$, $c_2=2$, $n_1=46$, $n_2=73$, an AFI(p_1)=0.1863. Dodge and Romig give the optimal sampling plan as $c_1=0$, $c_2=2$, $n_1=55$, $n_2=60$.

TABLE III
Comparisons of Computed Results with Other Standard Plans.

Parameters	Plans using minimum AFI criterion	Plans using Minimum ASN criterion	Dodge-Romig plans (Min AFI)	MIL-STD-105D Normal-II	MIL-STD-105D Tightened-II
AOQL=.01 $\alpha = .0347$ $\beta = .0736$ $p_1 = .0063$ $p_2 = .033$ $N=350$	$c_1=0$ $c_2=2$ $n_1=43$ $n_2=79$ AFI=.2005	$c_1=0$ $c_2=2$ $n_1=36$ $n_2=58$	$c_1=0$ $c_2=2$ $n_1=55$ $n_2=60$ $N=301-400$ AFI=.2321	$c_1=0$ $c_2=1$ $r_1=2$ $r_2=2$ $n_1=30$ $n_2=50$ AFI=.2396	$c_1=0$ $c_2=1$ $r_1=2$ $r_2=2$ $n_1=80$ $n_2=80$ AFI=.4421
AOQL=.02 $\alpha = .0381$ $\beta = .2029$ $p_1 = .015$ $p_2 = .067$ $N=350$	$c_1=0$ $c_2=3$ $n_1=26$ $n_2=35$ AFI=.1465	$c_1=0$ $c_2=3$ $n_1=35$ $n_2=52$	$c_1=0$ $c_2=3$ $n_1=33$ $n_2=53$ $N=301-400$ AFI=.184522	$c_1=0$ $c_2=3$ $r_1=3$ $r_2=4$ $n_1=32$ $n_2=32$ AFI=.1387	$c_1=0$ $c_2=1$ $r_1=2$ $r_2=2$ $n_1=32$ $n_2=32$ AFI=.2885
AOQL=.025 $\alpha = .0389$ $\beta = .0844$ $p_1 = .023$ $p_2 = .09$ $N=500$	$c_1=1$ $c_2=6$ $n_1=42$ $n_2=76$ AFI=.145115	$c_1=1$ $c_2=6$ $n_1=50$ $n_2=80$	$c_1=1$ $c_2=6$ $n_1=50$ $n_2=80$ $N=401-500$ AFI=.1858	$c_1=1$ $c_2=4$ $r_1=4$ $r_2=5$ $n_1=32$ $n_2=32$ AFI=.0924	$c_1=0$ $c_2=3$ $r_1=3$ $r_2=4$ $n_1=32$ $n_2=32$ AFI=.1630
AOQL=.03 $\alpha = .0433$ $\beta = .1062$ $p_1 = .04$ $p_2 = .165$ $N=450$	$c_1=0$ $c_2=3$ $N_1=11$ $n_2=46$ AFI=.078748	$c_1=1$ $c_2=4$ $N_1=25$ $n_2=23$	$c_1=1$ $c_2=4$ $N_1=16$ $n_2=34$ $N=401-500$ AFI=.0700	$c_1=2$ $c_2=6$ $r_1=5$ $r_2=7$ $n_1=32$ $n_2=32$ AFI=.0904	$c_1=1$ $c_2=4$ $r_1=4$ $R_2=3$ $n_1=32$ $n_2=32$ AFI=.1803
AOQL=.05 $\alpha = .0442$ $\beta = .0650$ $p_1 = .04$ $p_2 = .18$ $N=1500$	$c_1=1$ $c_2=11$ $n_1=22$ $n_2=111$ AFI=.03524	$c_1=1$ $c_2=4$ $n_1=26$ $n_2=24$	$c_1=0$ $c_2=4$ $n_1=17$ $n_2=33$ $N=1001-2000$ AFI=.0651	$c_1=5$ $c_2=12$ $r_1=9$ $r_2=13$ $n_1=80$ $n_2=80$ AFI=.0673	$c_1=3$ $c_2=11$ $r_1=7$ $r_2=12$ $n_1=80$ $n_2=80$ AFI=.0982
AOQL=.05 $\alpha = .0405$ $\beta = .0987$ $p_1 = .04$ $p_2 = .17$ $N=350$	$c_1=0$ $c_2=3$ $n_1=11$ $n_2=44$ AFI=.091434	$c_1=1$ $c_2=4$ $n_1=23$ $n_2=24$	$c_1=0$ $c_2=4$ $n_1=16$ $n_2=33$ $N=301-400$ AFI=.125809	$c_1=2$ $c_2=6$ $r_1=3$ $r_2=7$ $n_1=32$ $n_2=32$ AFI=.113002	$c_1=1$ $c_2=4$ $r_1=4$ $r_2=5$ $n_1=32$ $n_2=32$ AFI=.2043

The computer results for a 2% AOQL with $\alpha=0.0478$, $p_1=0.02$ and $N=1500$ follow:

c1	c2	n1	n2	AFI(p1)
0	7	28	128	.095858
1	7	54	144	.093983
2	7	81	116	.101264
0	8	28	193	.096234
1	8	55	166	.092043
2	8	81	142	.098126
0	9	28	217	.097668
1	9	55	190	.091320
2	9	82	163	.096105
0	10	28	242	.101074
1	10	55	214	.092049
2	10	82	187	.095498

Dodge and Romig give the optimal plan of $c_1=1$, $c_2=8$, $n_1=80$, $n_2=160$ with the AFI(p1) being 0.1444. The minimum sampling plan found above ($c_1^*=1$, $c_2^*=9$, $n_1^*=55$, $n_2^*=190$) has a lower AFI(p1) value than that corresponding to $c_1=1$, $c_2=8$. The difference is small; however this may account for some of the discrepancies in the results.

The sampling plans derived from large lot sizes do not correspond well with the Dodge-Romig plans. For the parameters $AOQL=0.05$, $\alpha=0.0442$, $p_1=0.04$ and $N=1500$,

AFI based algorithm plans	Dodge-Romig AOQL plans	MIL-STD-105D Normal-II	MIL-STD-105D Tightened-II
$c_1=1$	$c_1=0$	$c_1=5$	$c_1=3$
$c_2=11$	$c_2=4$	$c_2=12$	$c_2=11$
$n_1=22$	$n_1=17$	$r_1=9$	$R_1=7$
$n_2=111$	$n_2=33$	$r_2=13$	$R_2=12$
		$n_1=80$	$n_1=80$
		$b_2=80$	$n_2=80$

The c^* value of the single sampling plan based on the AFI criterion is 5. Since c_2 is initially set to c^* , the acceptance numbers found by the AFI algorithm and those of the Dodge-Romig plans would never be the same. It appears that for large values of N , the minimum AFI plans resulting from the described program more closely resemble those of the MIL-STD-105D plans.

The results using the minimum AFI(p_1) criterion are very similar to those using a minimum ASN criterion. The AFI based plans and the Dodge-Romig plans deal with rectifying inspection whereas the minimum ASN based plans do not.

In Table III, note that the total number sampled, n_1+n_2 , is very close in the AFI algorithm, ASN algorithm, and Dodge-Romig plans. The individual values of n_1 and n_2 vary to a larger degree than the total. The n_1 value from the ASN algorithm plan is greater than that of the Dodge-Romig plans and n_2 is less than the n_2 value of the Dodge-Romig plans. Also, $n_1(\text{AFI})$ is less than $n_1(\text{Dodge-Romig})$ and $n_2(\text{AFI})$ is greater than $n_2(\text{Dodge-Romig})$.

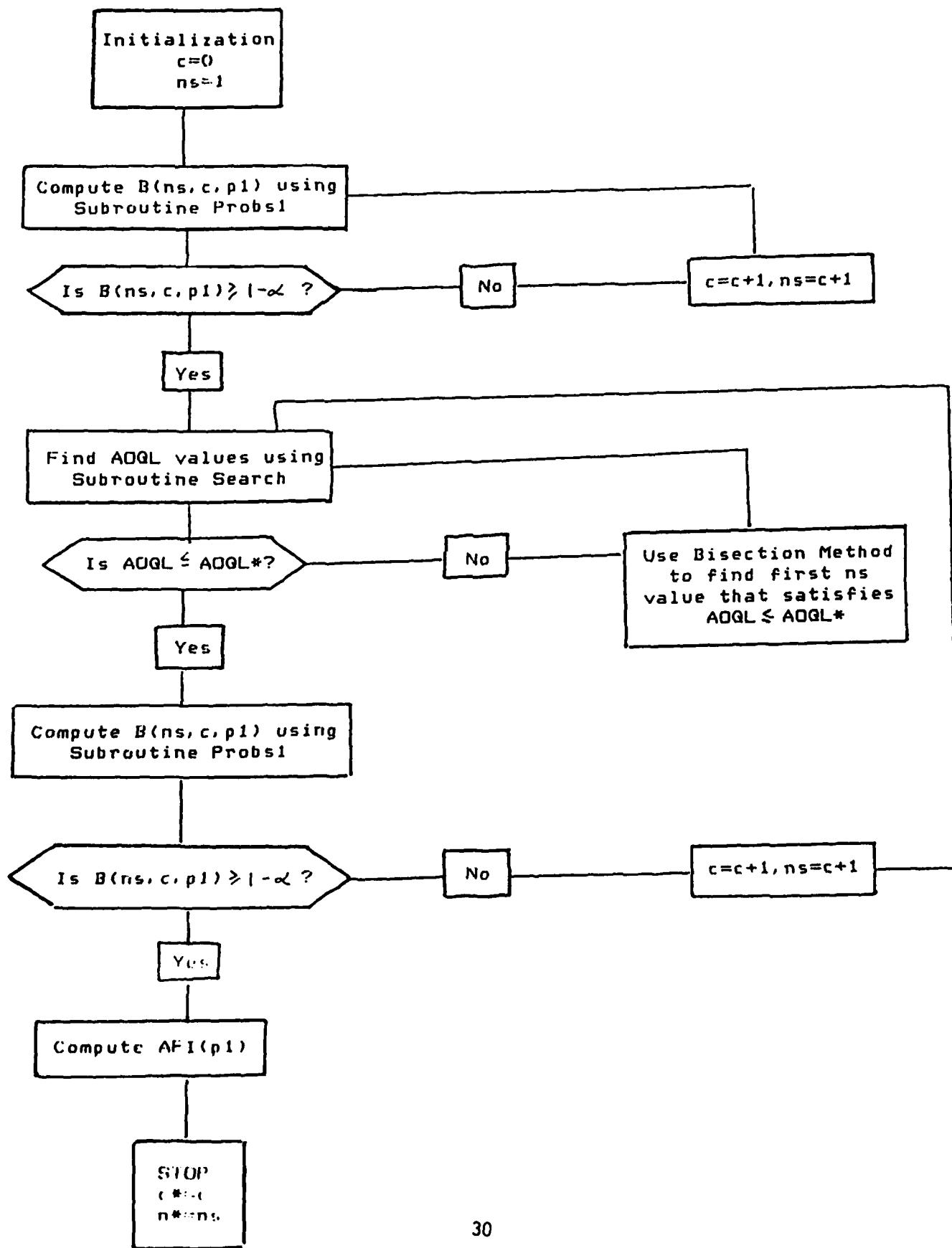
The MIL-STD-105D tables specify a fixed sample for a given lot size. In most cases, the total number sampled does not correspond well to the other plans. MIL-STD Level II plans for Normal and Tightened inspection are included in the table. It is not clear which specifications most closely related to the other sampling plans. In some instances the Normal inspection plans resemble the other plans, and in other cases the Tightened inspection plans are more similar.

The computer based technique is advantageous in that plans corresponding to specific AOQL values, lot sizes, and process averages can be found directly rather than relying on ranges of lot size and process averages which can yield only approximate optimal plans.

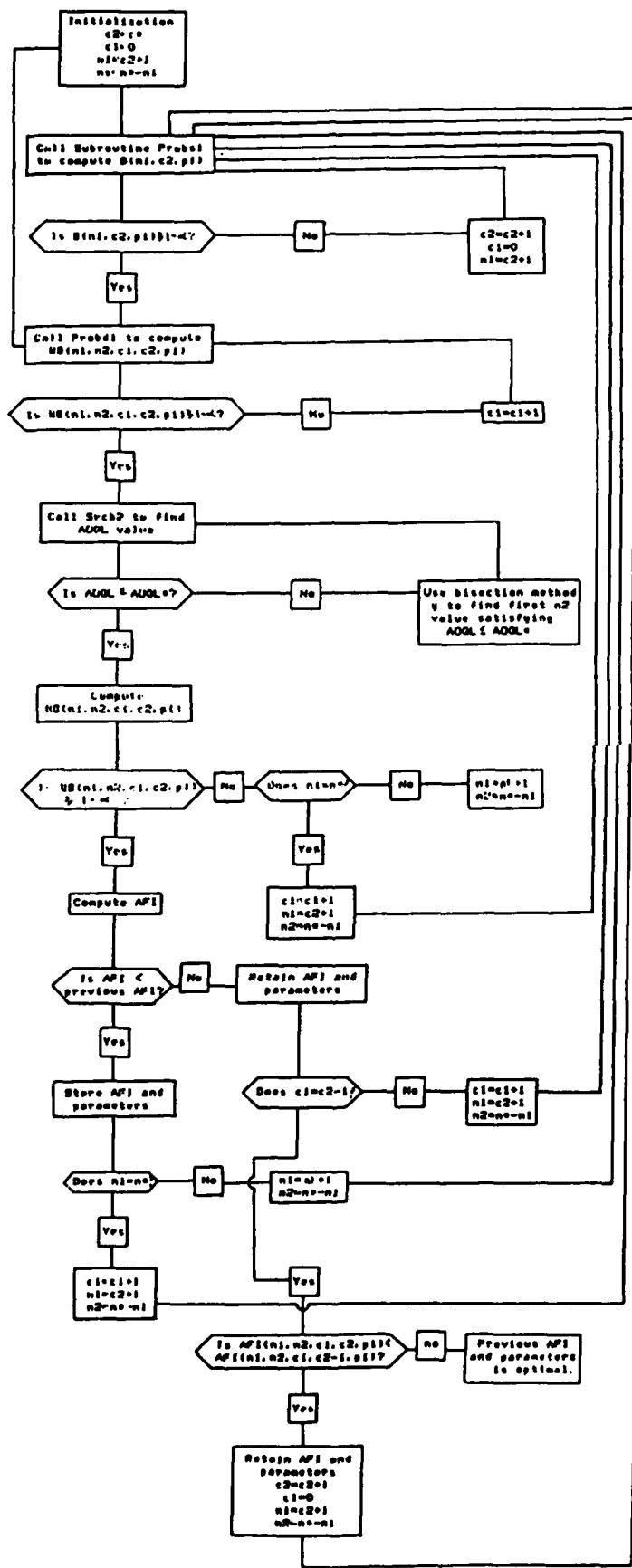
REFERENCES

1. Dodge, H. F. and Romig, H. G., Sampling Inspection Tables, John Wiley & Sons, Inc., 1959.
2. Grant, E. L. and Leavenworth, R. S. Statistical Quality Control, McGraw-Hill Book Co., 5th Edition, 1980.
3. Wagner, H. W., Principles of Operations Research, Prentice-Hall, Inc. 1969.

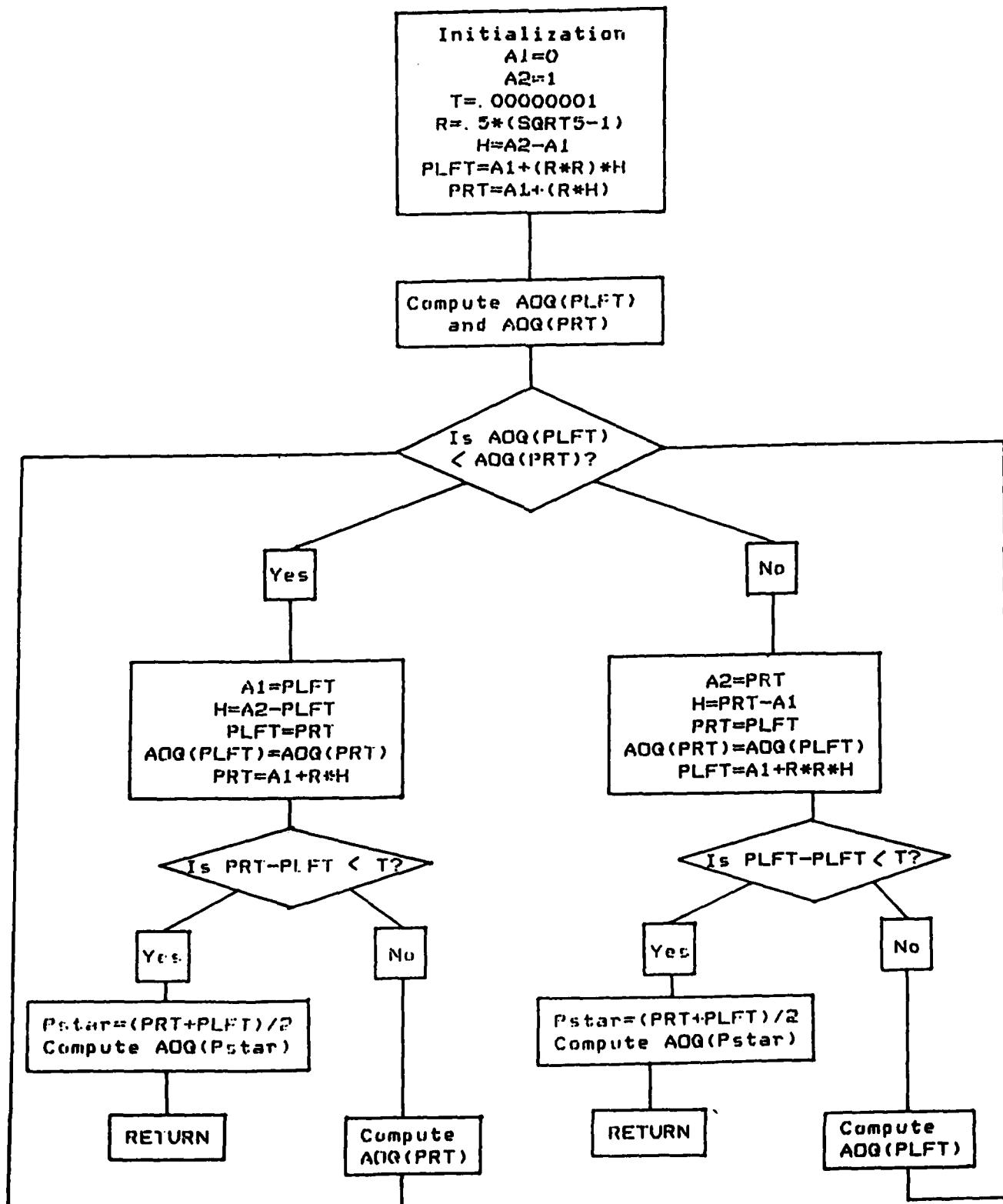
APPENDIX
SINGLE SAMPLING FLOW CHART



DOUBLE SAMPLING FLOW CHARTS



SUBROUTINE SEARCH



QAFI.FOR

```

0001 C
0002 C***** QUALITY CONTROL PROGRAM TO DERIVE DOUBLE SAMPLING
0003 C PLANS TO MINIMIZE AVERAGE FRACTION INSPECTED.
0004 C
0005 C
0006 C PROGRAMMED BY JO ELLEN WALKER
0007 C DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
0008 C UNIVERSITY OF FLORIDA
0009 C GAINESVILLE, FLORIDA 32611
0010 C*****+
0011 C
0012 C SUBROUTINE QAFI
0013 C INTEGER CJ, C2, C, CSTAR, C2M1
0014 C INTEGER RJ, RIMJ
0015 C DOUBLE PRECISION SUMLOG
0016 C BYTE OUTFILE(8)
0017 C COMMON/BLK4/ALPHA, BETA
0018 C COMMON/BLK6/C1, C2
0019 C COMMON/BLK7/SUMLOG(4000)
0020 C COMMON/BLK8/N
0021 C COMMON/BLK12/OUTFILE
0022 C*****+
0023 C INPUT PARAMETERS
0024 C*****+
0025 C KINDEX=1
0026 C
0027 C 10 FORMAT(10X, 'ENTER VALUE OF ALPHA')
0028 C READ(5, *)ALPHA
0029 C WRITE(5, 15)
0030 C 15 FORMAT(10X, 'ENTER VALUE OF PO')
0031 C READ(5, *)PO
0032 C WRITE(5, 20)
0033 C 20 FORMAT(10X, 'ENTER AOQL VALUE')
0034 C READ(5, *)AOQL
0035 C WRITE(5, 25)
0036 C 25 FORMAT(10X, 'ENTER LOT SIZE')
0037 C READ(5, *)NNN
0038 C
0039 C
0040 C
0041 C
0042 C 28 FORMAT('1', //10X, 'DEPT. OF ISE '
0043 C '$/, 10X, 'UNIVERSITY OF FLORIDA '
0044 C '$/5X, 5(*'), 'DOUBLE SAMPLING SYSTEM TO MINIMIZE AFI',
0045 C $5(')'), 2X, /)
0046 C
0047 C
0048 C 30 FORMAT(10X, 'ALPHA= ', 2X, F8.6)
0049 C
0050 C
0051 C 35 FORMAT(10X, 'PO= ', 2X, F8.6)
0052 C
0053 C
0054 C 40 FORMAT(10X, 'AOQL= ', 2X, F8.6)
0055 C
0056 C
0057 C 45 FORMAT(10X, 'N= ', 2X, I6)

```

QAFI

```

0058      ****
0059      C      COMPUTE SINGLE SAMPLING PLAN
0060      ****
0061      C
0062      C      INITIALIZATION
0063      C
0064      ****
0065      NS=1
0066      C=0
0067      N=0
0068      AFI=1.D0
0069      C
0070      WRITE(5,50)
0071      WRITE(1,50)
0072      50 FORMAT(//10X, '*** SINGLE SAMPLING PLAN ***')
0073      ****
0074      C      FIND NS, C COMBO THAT SATISFIES L(PO) G.T. 1-ALPHA
0075      ****
0076      55 CALL PROBS1(NS, PO, C, BXLEC)
0077      IF(BXLEC.LT.(1.D0-ALPHA))C=C+1
0078      IF(BXLEC.LT.(1.D0-ALPHA))NS=C+1
0079      ****
0080      C      SEARCH TO FIND MIN NS VALUE SUCH THAT AOQL L.T. AOQL*
0081      ****
0082      60 CALL SEARCH(NNN, C, NS, SAOQ)
0083      IF(SAOQ.LE.AOQL)GOTO 75
0084      NSTEMP=NNN
0085      CALL SEARCH(NNN, C, NSTEMP, SAOQ)
0086      IF(SAOQ.GT.AOQL)C=C+1
0087      C
0088      C      IF(SAOQ.GT.AOQL)NS=C+1
0089      C
0090      C      IF(SAOQ.GT.AOQL)NS=C+1
0091      C
0092      BL=NS
0093      BH=NNN
0094      65 NSTEMP=IIDINT((BL+BH)/2.D0)
0095      CALL SEARCH(NNN, C, NSTEMP, SAOQ)
0096      IF(SAOQ.LE.AOQL)BH=NSTEMP
0097      C
0098      IF(SAOQ.LE.AOQL)AOQ=SAOQ
0099      C
0100      IF(SAOQ.GT.AOQL)BL=NSTEMP
0101      IF((BH-BL).EQ.1.D0)GOTO 70
0102      GOTO 65
0103      70 NS=NSTEMP
0104      IF(SAOQ.GT.AOQL)NS=BH
0105      ****
0106      C      CHECK THAT NS, C COMBO STILL SATISFIES L(PO) CONSTRAINT
0107      ****
0108      75 CALL PROBS1(NS, PO, C, BXLEC)
0109      IF(BXLEC.GE.(1.D0-ALPHA))GOTO 80
0110      C      C=C+1
0111      C      NS=C+1
0112      C
0113      C      GOTO 60
0114

```

QAFI

```
0115 C*****  
0116 C COMPUTE AFI  
0117 C*****  
0118 80 ATIPO=NS*BXI EC HNNN*(1 DO-BXLEC)  
0119 AFIPO=ATIPO/NNN  
0120 C  
0121 NSTAR=NS  
0122 C  
0123 WRITE(5,85)NS  
0124 WRITE(1,85)NS  
0125 85 FORMAT(//10X, 'NS=', I3)  
0126 WRITE(5,90)C  
0127 WRITE(1,90)C  
0128 90 FORMAT(10X, 'C=', I2)  
0129 WRITE(5,95)AFIPO  
0130 WRITE(1,95)AFIPO  
0131 95 FORMAT(10X, 'AFI(PO)=', F8.6)  
0132 C  
0133 100 CSTAR=C  
0134 C*****  
0135 DOUBLE SAMPLING  
0136 C*****  
0137 C  
0138 WRITE(5,105)  
0139 WRITE(1,105)  
0140 105 FORMAT(//10X, '** DOUBLE SAMPLING PLANS **')  
0141 PLAN=1.D0  
0142 C2=CSTAR  
0143 110 R1=C2+1  
0144 C2M1=C2-1  
0145 C1PO=C1+1  
0146 R1M1=R1-1  
0147 DDATI=NNN  
0148 ITCMIN=1.  
0149 C  
0150 C1=0  
0151 JJ=0  
0152 C  
0153 WRITE(5,115)  
0154 WRITE(1,115)  
0155 115 FORMAT(//10X, 'C1', 6X, 'C2', 7X, 'N1', BX, 'N2', 9X, 'AFI', //)  
0156 C*****  
0157 CALCULATE FIRST SAMPLE NUMBER  
0158 C  
0159 FROM RESULTS OF PREVIOUS RUNS, IT WAS FOUND THAT N1 IS NOT LESS  
0160 THAN NSTAR/8. THUS, THE INITIAL VALUE OF N1 IS SET ACCORDINGLY.  
0161 C*****  
0162 C  
0163 120 DO 165 LL=INT(NSTAR/8),NSTAR  
0164 N1=LL  
0165 IF(N1.LT.C2)N1=R1M1  
0166 C*****  
0167 C CHECK B(N1,PO,C2) G.T. 1-ALPHA CONSTRAINT  
0168 C*****  
0169 125 CALL PROBS1(N1,PO,C2,BXLEC)  
0170 IF(BXLEC.LT.(1.-ALPHA))GOTO 175  
0171 C*****
```

QAF1

```
0172 C      CALCULATE SECOND SAMPLE
0173 C*****N2=NSTAR-N1*****
0174 C*****CHECK THAT DOUBLE PROBABILITY G.T. 1-ALPHA
0175 C*****C1=C1+1
0176 C*****GOTO 130
0177 C*****130 CALL PROBD1(N1,N2,PO,DPROB,KINDEX,R1)
0178 C*****IF(DPROB.GE.(1.-ALPHA))GOTO 135
0179 C*****IF(C1.EQ.C2M1)GOTO 175
0180 C*****C1=C1+1
0181 C*****GOTO 130
0182 C*****135 CALL SRCH2(NNN,N1,N2,AQQL)
0183 C*****IF(AQQL.LE.AQQL)GOTO 150
0184 C*****N2 WILL NOT BE LESS THAN N1*9, THE INITIAL LOWER BOUND ON N2.
0185 C*****N2TEMP=N1*9
0186 C*****CALL SRCH2(NNN,N1,N2TEMP,AQQL)
0187 C*****IF(AQQL.GT.AQQL)GOTO 165
0188 C*****BL=N2
0189 C*****BH=N1*9
0190 C*****140 N2TEMP=INT((BL+BH)/2)
0191 C*****CALL SRCH2(NNN,N1,N2TEMP,AQQL)
0192 C*****IF(AQQL.LE.AQQL)BH=N2TEMP
0193 C*****IF(AQQL.LE.AQQL)FAOQ=AQQL
0194 C*****IF(AQQL.GT.AQQL)BL=N2TEMP
0195 C*****IF((BH-BL).EQ.1.)GOTO 145
0196 C*****GOTO 140
0197 C*****145 N2=N2TEMP
0198 C*****IF(AQQL.GT.AQQL)N2=BH
0199 C*****CHECK THAT BINOMIAL PROBABILITIES ARE G.T. 1-ALPHA
0200 C*****CALL PROBD1(N1,N2,PO,DPROB,KINDEX,R1)
0201 C*****IF(DPROB.LT.(1.-ALPHA))GOTO 165
0202 C*****150 CALL PROBS1(N1,PO,C1,BXLEC)
0203 C*****PA2=DPROB-BXLEC
0204 C*****COMPUTE ATI
0205 C*****DATI=N1*DPROB+N2*PA2+NNN*(1.-DPROB)
0206 C*****IF THE ATI INCREASES, CONTINUE FOR 5 ADDITIONAL INCREASING
0207 C*****ITERATIONS. THEN INCREMENT C1 AND CONTINUE.
0208 C*****IF(JJ.EQ.4)GOTO 155
0209 C*****IF(DATI.GE.DDATI)JJ=JJ+1
0210 C*****IF(DATI.GE.DDATI)GOTO 165
0211 C*****DDATI=DATI
0212 C*****DDAOQ=FAOQ
0213 C*****C
```

QAF I

```

0229      K1=C1
0230      K2=C2
0231      K3=N1
0232      K4=N2
0233      C
0234      GOTO 165
0235      *****
0236      C      MINIMUM OF CELL (TCMIN) FOUND
0237      *****
0238      155 TCMIN=DDAT1/NNN
0239      DDAT1=NNN
0240      C
0241      WRITE(5, 160)K1, K2, K3, K4, TCMIN
0242      WRITE(1, 160)K1, K2, K3, K4, TCMIN
0243      160 FORMAT(10X, I2, 6X, I2, 5X, I4, 6X, I4, 7X, F8.6)
0244      *****
0245      C      IF MINIMUM OF COLUMN IS FOUND, INCREASE C2
0246      *****
0247      IF(TCMIN.GE.TTCMIN)GOTO 170
0248      KK1=K1
0249      KK2=K2
0250      KK3=K3
0251      KK4=K4
0252      TTCMIN=TCMIN
0253      TTAOQ=DDAOQ
0254      C
0255      IF (C1.EQ.C2M1) GOTO 175
0256      C1=C1+1
0257      C
0258      JJ=0
0259      N1=KK3
0260      C
0261      GOTO 125
0262      C
0263      165 CONTINUE
0264      *****
0265      C      MINIMUM OF COLUMN (TMIN) FOUND
0266      *****
0267      170 TMIN=TTCMIN
0268      TAOQ=TTAOQ
0269      *****
0270      C      IF MINIMUM SAMPLING PLAN FOUND, STOP
0271      *****
0272      IF(TMIN.GE.PLAN)GOTO 190
0273      C
0274      PLAN=TMIN
0275      KKK1=KK1
0276      KKK2=KK2
0277      KKK3=KK3
0278      KKK4=KK4
0279      175 C2=C2+1
0280      C
0281      C      NEW BOUNDS ON SAMPLING PLAN CALCULATED FOR NEW VALUE OF C2
0282      *****
0283      180 CALL SEARCH(NNN, C2, NSTAR, AOQ)
0284      IF(AOQ.LE.AOGL)GOTO 185
0285      IF(NSTAR.GT.NNN)GOTO 175

```

QAF I

```
0286      NSTAR=NSTAR+1
0287      GOTO 180
0288      C
0289      C
0290      185 CALL PROBS1(NSTAR, P0, C2, BXLEC)
0291      GOTO 110
0292      C
0293      C
0294      190 WRITE(5, 195)
0295      WRITE(1, 195)
0296      195 FORMAT(//10X, 'SAMPLING PLAN MINIMUMS')
0297      WRITE(5, 200)KKK1, KKK2
0298      WRITE(1, 200)KKK1, KKK2
0299      200 FORMAT(/10X, 'C1=', I2, 2X, 'C2=', I2)
0300      WRITE(5, 205)KKK3, KKK4
0301      WRITE(1, 205)KKK3, KKK4
0302      205 FORMAT(10X, 'N1=', I3, 2X, 'N2=', I3)
0303      WRITE(5, 210)PLAN
0304      WRITE(1, 210)PLAN
0305      210 FORMAT(10X, 'MINIMUM AFI=', F8.6)
0306      215 RETURN
0307      END
```

```

0001      SUBROUTINE SEARCH(NNN, C, NS, AQQ)
0002      *****
0003      C      SEARCH TO FIND VALUE OF PSTAR USING GOLDEN
0004      C      SECTION METHOD. INITIAL LIMITS OF 0 AND 1
0005      *****
0006      INTEGER C
0007      DOUBLE PRECISION SUMLOG
0008      COMMON/B1 K7/SUMLOG(4000)
0009      COMMON/B1 K8/N
0010      C
0011      A1=0. DO
0012      A2=1. DO
0013      T=1. D-3
0014      R=5. D-1*(DSQRT(5. DO)-1. DO)
0015      H=A2-A1
0016      PLFT=A1+(R*R)*H
0017      PRT=A1+(R*T)
0018      C
0019      CALL PROBS1(NS, PLFT, C, BXLEC)
0020      ATI=(NS*BXLEC)+NNN*(1. DO-BXLEC)
0021      AFI1=ATI/NNN
0022      AQQ1=PLFT*(1. DO-AFI1)
0023      CALL PROBS1(NS, PRT, C, BXLEC)
0024      ATI=(NS*BXLEC)+NNN*(1. DO-BXLEC)
0025      AFI2=ATI/NNN
0026      AQQ2=PRT*(1. DO-AFI2)
0027      GOTO 110
0028      C
0029      C
0030      100 CALL PROBS1(NS, PLFT, C, BXLEC)
0031      ATI=(NS*BXLEC)+NNN*(1. DO-BXLEC)
0032      AFI1=ATI/NNN
0033      AQQ1=PLFT*(1. DO-AFI1)
0034      GO TO 110
0035      C
0036      105 CALL PROBS1(NS, PRT, C, BXLEC)
0037      ATI=(NS*BXLEC)+NNN*(1. DO-BXLEC)
0038      AFI2=ATI/NNN
0039      AQQ2=PRT*(1. DO-AFI2)
0040      C
0041      110 IF(AQQ1.LT.AQQ2) GOTO 115
0042      A2=PRT
0043      H=PRT-A1
0044      IF(ABS(PRT-PLFT).LE.T)GOTO 120
0045      C
0046      PRT=PLFT
0047      PLFT=A1+(R*R)*H
0048      AQQ2=AQQ1
0049      GO TO 100
0050      115 A1=PLFT
0051      H=A2-PLFT
0052      C
0053      IF(ABS(PRT-PLFT).LE.T)GOTO 120
0054      PLFT=PRT
0055      PRT=A1+R*T
0056      AQQ1=AQQ2
0057

```

SEARCH

```
0058      GO TO 105
0059      C
0060      C
0061      120 PS=(PLF1(PRT)/2,DO
0062          CALL PROBS1(NS,PS,C,BXLEC)
0063          AFI=((NS*BXLEC)+NNN*(1,DO-BXLEC))/NNN
0064          AQQ=PS*(1,DO-AF1)
0065          RETURN
0066          END
```

```

0001      C      SUBROUTINE SRCH2(NNN,N1,N2,AQ)
0002      C
0003      C      INTEGER C1,C2,R1
0004      C      COMMON/BLK6/C1,C2
0005      C
0006      C      KINDEX=1
0007      C      A1=0.
0008      C      A2=1.
0009      C      T=.0001
0010      C      R=.5*(DSQRT(.5,DO)-1.)
0011      C      H=A2-A1
0012      C      PLFT=A1+(R*R)*H
0013      C      PRT=A1+(R*H)
0014      C
0015      C
0016      C      CALL PROBS1(N1,PLFT,C1,PA1)
0017      C      CALL PROBD1(N1,N2,PLFT,DPROB,KINDEX,R1)
0018      C      PA2=DPROB-PA1
0019      C      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
0020      C      AFI1=ATI/NNN
0021      C      AQ1=PLFT*(1.-AFI1)
0022      C      CALL PROBS1(N1,PRT,C1,PA1)
0023      C      CALL PROBD1(N1,N2,PRT,DPROB,KINDEX,R1)
0024      C      PA2=DPROB-PA1
0025      C      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
0026      C      AFI2=ATI/NNN
0027      C      AQ2=PRT*(1.-AFI2)
0028      C      GOTO 110
0029      C
0030      C
0031      100 CALL PROBS1(N1,PLFT,C1,PA1)
0032      C      CALL PROBD1(N1,N2,PLFT,DPROB,KINDEX,R1)
0033      C      PA2=DPROB-PA1
0034      C      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
0035      C      AFI1=ATI/NNN
0036      C      AQ1=PLFT*(1.-AFI1)
0037      C      GOTO 110
0038      105 CALL PROBS1(N1,PRT,C1,PA1)
0039      C      CALL PROBD1(N1,N2,PRT,DPROB,KINDEX,R1)
0040      C      PA2=DPROB-PA1
0041      C      ATI=DPROB*N1+PA2*N2+NNN*(1.-DPROB)
0042      C      AFI2=ATI/NNN
0043      C      AQ2=PRT*(1.-AFI2)
0044      C
0045      C
0046      110 IF(AQ1.LT.AQ2)GOTO 115
0047      C
0048      C
0049      C      A2=PRT
0050      C      H=PRT-A1
0051      C      IF(ABS(PRT-PLFT).LE.T)GOTO 120
0052      C      PRT=PLFT
0053      C      PLFT=A1+(R*R)*H
0054      C      AQ2=AQ1
0055      C      GOTO 100
0056      115 A1=PLFT
0057      C

```

SRCHZ

0058 C
0059 C
0060 C
0061 C
0062 C
0063 C
0064 C
0065 C
0066 C
0067 C
0068 C
0069 C
0070 C
0071 C
0072 C
0073 C
0074 C

IF(A2=PLF1)
1F(ABS(PRT-PLF1),LE.,T)GOTO 120
PLF1=PRT
PRT=A1+R*BH
AOQ1=AOQ2
GOTO 105

120 PS=(PLF1+PRT)/2.
CALL PROBS1(N1,PS,C1,PA1)
CALL PROBD1(N1,N2,PS,DPROB,KINDEX,R1)
PA2=DPROB-PA1
AFI=(DPROB*N1+PA2*N2+NNN*(1.-DPROB))/NNN
AOQ=PS*(1.-AFI)
RETURN
END

```

0001      SUBROUTINE PRBRS1(NN, P, C, BXLEC)
0002      C*****THIS SUBROUTINE COMPUTES CUMULATIVE BINOMIAL
0003      C PROBABILITIES
0004      C*****INTEGER C
0005      C      DOUBLE PRECISION SUMLOG
0006      C
0007      COMMON/B1 K7/SUMI LOG(4000)
0008      COMMON/B1 K3/N
0009      C
0010      C      Q=J.-P
0011      C*****BINOMIAL PROB. WHEN C=0
0012      C*****CSUMS=Q*NN
0013      IF (C, EQ, 0) GOTO 45
0014      C*****AVOID RECOMPUTING SUMLOG(I)'S ALREADY IN MEMORY
0015      C*****IF (N-NN) 10, 25, 25
0016      10 M=N+1
0017      C*****COMPUTE N SUMLOGS-EQUIVALENT TO N-FACTORIAL
0018      C*****IF (M.GT.1) GOTO 15
0019      SUMI LOG(1)=0,
0020      IF (NN.LE.1) GOTO 25
0021      M=?
0022      15 DO 20 I=M, NN
0023          SUMLOG(I)=DLOG10(DFLOAT(I))+SUMLOG(I-1)
0024          20 CONTINUE
0025      C*****COMPUTE C SUMS-EQUIVALENT TO SSUM OF PROB. COMPIN.
0026      C      I.E. CUMULATIVE BINOMIAL DISTRIBUTION COMPUTATION
0027      C*****25 IF(NN.GT.N) N=NN
0028      C*****DETERMINE BEST NUMBER HANDLING LOOP
0029      C*****IF (NN.GT.300) GOTO 35
0030      DO 30 K=1,C
0031          CSUMS=10.**(SUMLOG(NN)-SUMLOG(NN-K)-SUMLOG(K))
0032          1          *P**K*Q**((NN-K)+CSUMS
0033          30 CONTINUE
0034          GOTO 45
0035      C*****LOOP FOR LARGE EXPONENTS
0036      C*****35 DO 40 K=1,C
0037          CSUMS=10.**(SUMLOG(NN)-SUMLOG(NN-K)-SUMLOG(K)
0038          1          +K*DLOG10(DBLE(P))+(NN-K)*DLOG10(DBLE(Q))+CSUMS
0039          40 CONTINUE
0040      C
0041          45 BXLEC = CSUMS
0042          RETURN
0043          END

```

```

0001      SUBROUTINE PROBD1(N1,N2,P,DPROB,K,R1)
0002  ****
0003  C      THIS SUBROUTINE COMPUTES DOUBLE PROBABILITIES FOR
0004  C      COMPUTING SECOND SAMPLE NUMBER OF DOUBLE SAMPLING NUMBER
0005  C  ****
0006  COMMON/BLK6/C1,C2
0007  INTEGER C1,C2,R1
0008  C
0009  IF(K.EQ.1) CALL PROBS1(N1,P,C1,BXLEC)
0010  IF(K.EQ.2) CALL PROBS2(N1,P,C1,BXLEC)
0011  DPROB=BXLEC
0012  TEMP=BXLEC
0013  NTEMP=C1+1
0014  KTEMP=R1-1
0015  DO 10 IX=NTEMP, KTEMP
0016    I=IX
0017    J=C2-I
0018    IF(K.EQ.1) CALL PROBS1(N1,P,I,BXLEC)
0019    IF(K.EQ.2) CALL PROBS2(N1,P,I,BXLEC)
0020    PROB1=BXLEC-TEMP
0021    TEMP=BXLEC
0022    IF(K.EQ.1) CALL PROBS1(N2,P,J,BXLEC)
0023    IF(K.EQ.2) CALL PROBS2(N2,P,J,BXLEC)
0024    DPROB=DPROB+(PROB1*BXLEC)
0025  10 CONTINUE
0026  C
0027  RETURN
0028  END

```

